

# Bayesian Regression Model

Jayapriya Nagarajan & Shrankhala Jain

February 21, 2018

```
###Objective
##To predict how much icecream should be kept as stock on a hot day at 35
degree centigrade, such that the probability of run out of icecream is 2.5%

#Data considered for analyses
temp <- c(11.9,14.2,15.2,16.4,17.2,18.1,18.5,19.4,22.1,22.6,23.4,25.1)
units <- c(185L,215L,332L,325L,408L,421L,406L,412L,522L,445L,544L,614L)
log_units <- log(units)
n <- length(units)
market.size <- rep(800, n)
df <- data.frame(temp,units,log_units,market.size)

# Using brms package to derive 97.5% prediction credible interval
library(brms)

## Warning: package 'brms' was built under R version 3.4.3

## Loading required package: Rcpp

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.3

## Loading 'brms' package (version 2.1.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
## Run theme_set(theme_default()) to use the default bayesplot theme.

library(rstan)

## Warning: package 'rstan' was built under R version 3.4.3

## Loading required package: StanHeaders

## Warning: package 'StanHeaders' was built under R version 3.4.3

## rstan (Version 2.17.3, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend
calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```

# Linear Gaussian model
lin.mod <- brm(units ~ temp, data = df, family="gaussian")

## Compiling the C++ model

## Start sampling

##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 1).
##
## Gradient evaluation took 0.001 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 10
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.222 seconds (Warm-up)
##               0.117 seconds (Sampling)
##               0.339 seconds (Total)
##
##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)

```

```
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.272 seconds (Warm-up)
##               0.097 seconds (Sampling)
##               0.369 seconds (Total)
##
##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.2 seconds (Warm-up)
##               0.082 seconds (Sampling)
##               0.282 seconds (Total)
##
##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
```

```

## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.186 seconds (Warm-up)
##               0.111 seconds (Sampling)
##               0.297 seconds (Total)

# Log-transformed Linear Gaussian model
log.lin.mod <- brm(log_units ~ temp, data = df, family="gaussian")

## Compiling the C++ model
## Start sampling

##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 1).
##
## Gradient evaluation took 0.001 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 10
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:  200 / 2000 [ 10%] (Warmup)
## Iteration:  400 / 2000 [ 20%] (Warmup)
## Iteration:  600 / 2000 [ 30%] (Warmup)
## Iteration:  800 / 2000 [ 40%] (Warmup)
## Iteration: 1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.139 seconds (Warm-up)
##               0.114 seconds (Sampling)
##               0.253 seconds (Total)
##
##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!

```

```

##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration:  1200 / 2000 [ 60%] (Sampling)
## Iteration:  1400 / 2000 [ 70%] (Sampling)
## Iteration:  1600 / 2000 [ 80%] (Sampling)
## Iteration:  1800 / 2000 [ 90%] (Sampling)
## Iteration:  2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.133 seconds (Warm-up)
##                0.126 seconds (Sampling)
##                0.259 seconds (Total)
##
##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration:  1200 / 2000 [ 60%] (Sampling)
## Iteration:  1400 / 2000 [ 70%] (Sampling)
## Iteration:  1600 / 2000 [ 80%] (Sampling)
## Iteration:  1800 / 2000 [ 90%] (Sampling)
## Iteration:  2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.129 seconds (Warm-up)
##                0.141 seconds (Sampling)
##                0.27 seconds (Total)
##
##
## SAMPLING FOR MODEL 'gaussian brms-model' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0

```

```

seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.194 seconds (Warm-up)
##               0.158 seconds (Sampling)
##               0.352 seconds (Total)

# Poisson model
pois.mod <- brm(units ~ temp, data = df, family="poisson")

## Compiling the C++ model
## Start sampling

##
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.064 seconds (Warm-up)

```

```
##           0.057 seconds (Sampling)
##           0.121 seconds (Total)
##
##
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.054 seconds (Warm-up)
##              0.057 seconds (Sampling)
##              0.111 seconds (Total)
##
##
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
```

```

##
## Elapsed Time: 0.065 seconds (Warm-up)
##                0.059 seconds (Sampling)
##                0.124 seconds (Total)
##
##
## SAMPLING FOR MODEL 'poisson brms-model' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [ 0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.053 seconds (Warm-up)
##                0.052 seconds (Sampling)
##                0.105 seconds (Total)
##
# Binomial model
bin.mod <- brm(units | trials(market.size) ~ temp,data = df,
family="binomial")

## Compiling the C++ model
## Start sampling

##
## SAMPLING FOR MODEL 'binomial brms-model' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [ 0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)

```



```

## Iteration: 600 / 2000 [ 30%] (Warmup)
## Iteration: 800 / 2000 [ 40%] (Warmup)
## Iteration: 1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.107 seconds (Warm-up)
##               0.154 seconds (Sampling)
##               0.261 seconds (Total)
##
##
## SAMPLING FOR MODEL 'binomial brms-model' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:   200 / 2000 [ 10%] (Warmup)
## Iteration:   400 / 2000 [ 20%] (Warmup)
## Iteration:   600 / 2000 [ 30%] (Warmup)
## Iteration:   800 / 2000 [ 40%] (Warmup)
## Iteration:  1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.143 seconds (Warm-up)
##               0.124 seconds (Sampling)
##               0.267 seconds (Total)
##
##
## SAMPLING FOR MODEL 'binomial brms-model' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)

```

```

## Iteration: 200 / 2000 [ 10%] (Warmup)
## Iteration: 400 / 2000 [ 20%] (Warmup)
## Iteration: 600 / 2000 [ 30%] (Warmup)
## Iteration: 800 / 2000 [ 40%] (Warmup)
## Iteration: 1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.103 seconds (Warm-up)
##               0.142 seconds (Sampling)
##               0.245 seconds (Total)
##
##
## SAMPLING FOR MODEL 'binomial brms-model' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0
seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%] (Warmup)
## Iteration:  200 / 2000 [ 10%] (Warmup)
## Iteration:  400 / 2000 [ 20%] (Warmup)
## Iteration:  600 / 2000 [ 30%] (Warmup)
## Iteration:  800 / 2000 [ 40%] (Warmup)
## Iteration: 1000 / 2000 [ 50%] (Warmup)
## Iteration: 1001 / 2000 [ 50%] (Sampling)
## Iteration: 1200 / 2000 [ 60%] (Sampling)
## Iteration: 1400 / 2000 [ 70%] (Sampling)
## Iteration: 1600 / 2000 [ 80%] (Sampling)
## Iteration: 1800 / 2000 [ 90%] (Sampling)
## Iteration: 2000 / 2000 [100%] (Sampling)
##
## Elapsed Time: 0.156 seconds (Warm-up)
##               0.137 seconds (Sampling)
##               0.293 seconds (Total)
##
##Prediction Credible interval

modelData <- data.frame(
  Model=factor(c(rep("Linear model", 12), rep("Log-transformed LM",
12),rep("Poisson (log)",12),rep("Binomial (logit)",12)),levels=c("Linear
model","Log-transformed LM","Poisson (log)","Binomial (logit)"), ordered =
TRUE), Temperature=rep(temp, 4),Units_sold=rep(units,4),
rbind(predict(lin.mod),exp(predict(log.lin.mod) + 0.5 *

```

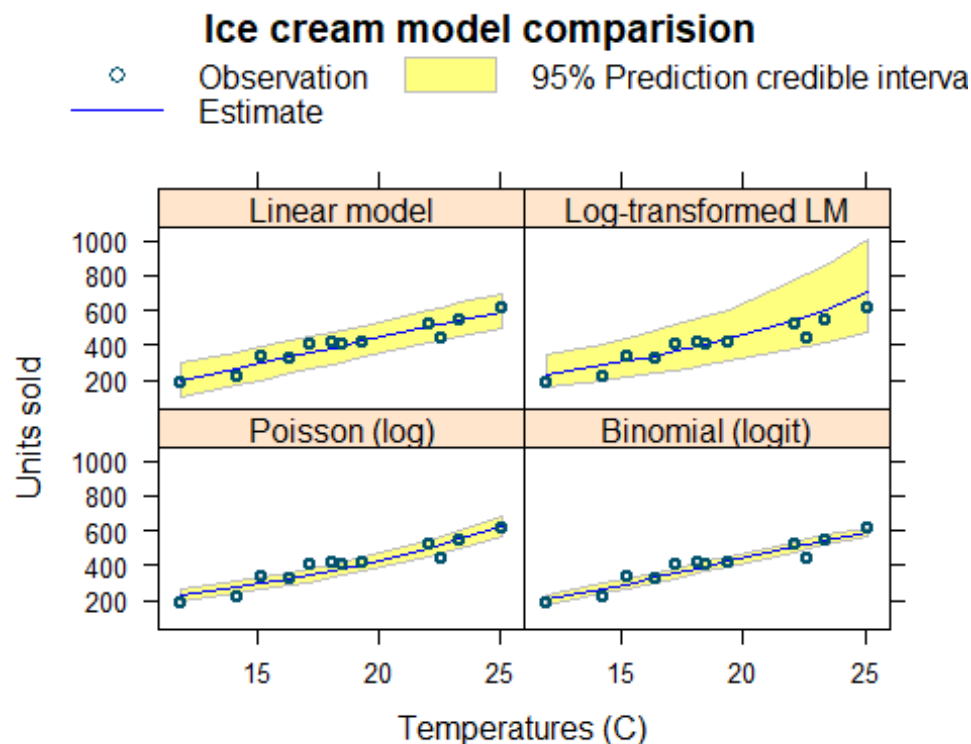
```
mean(extract(log.lin.mod$fit)[["sigma"]]))),predict(pois.mod),predict(bin.mod))
```

*#The predict function gives access to posterior predictive statistics, including the 97.5% prediction credible interval. Here going to combine the output of the models into dataframe to compare the prediction credible intervals of the four models in one chart.*

```
library(lattice)
```

```
# Plot the data
```

```
key <- list(
  rep=FALSE, lines=list(col=c("#00526D", "blue"), type=c("p","l"), pch=1),
  text=list(lab=c("Observation","Estimate")),rectangles =
list(col=adjustcolor("yellow", alpha.f=0.5),
border="grey"),text=list(lab="95% Prediction credible interval"))
xyplot(X2.5.ile + X97.5.ile + Estimate + Units_sold ~ Temperature | Model,
  data=modelData, as.table=TRUE, main="Ice cream model comparision",
  xlab="Temperatures (C)", ylab="Units sold",
  scales=list(alternating=1), key=key,
  panel=function(x, y){
    n <- length(x)
    k <- n/2
    upper <- y[(k/2+1):k]
    lower <- y[1:(k/2)]
    x <- x[1:(k/2)]
    panel.polygon(c(x, rev(x)), c(upper, rev(lower)),
      col = adjustcolor("yellow", alpha.f = 0.5),
      border = "grey")
    panel.lines(x, y[(k+1):(k+n/4)], col="blue")
    panel.points(x, y[(n*3/4+1):n], lwd=2, col="#00526D")
  })
```



*#The plot displays the posterior prediction credible intervals for the four models and it indicates over prediction of the Log-transformed linear model.*

```
## Stock to be maintained on a hot day
A <- function(samples){
  as.matrix(samples[,c("b_Intercept" ,"b_temp")])
}
x <- c(1, 35)
prob <- 0.975

#Linear model
lin.samples <- posterior_samples(lin.mod)
n <- nrow(lin.samples)
mu <- A(lin.samples) %*% x
sigma <- lin.samples[, "sigma"]
(lin.q <- quantile(rnorm(n, mu, sigma), prob))

##      97.5%
## 1040.875

#Log-transformed model
log.lin.samples <- posterior_samples(log.lin.mod)
mu <- A(log.lin.samples) %*% x
sigma <- log.lin.samples[, "sigma"]
(log.lin.q <- quantile(exp(rnorm(n, mu + 0.5*sigma^2, sigma)), prob))
```

```
## 97.5%
## 2493.38

#Poisson model
pois.samples <- posterior_samples(pois.mod)
mu <- exp(A(pois.samples) %*% x)
(pois.q <- quantile(rpois(n, mu) , prob))

## 97.5%
## 1510.025

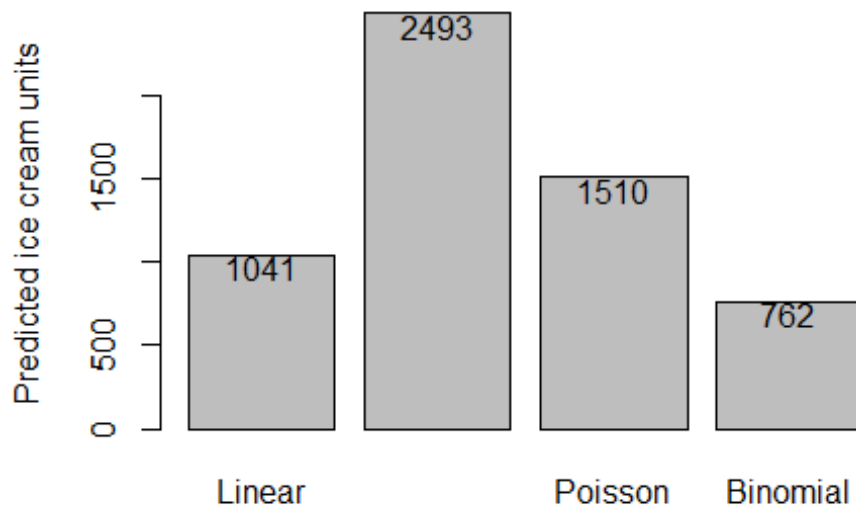
#Binomial model
bin.samples <- posterior_samples(bin.mod)
inv.logit <- function(u) exp(u)/(1+exp(u))
mu <- inv.logit( A(bin.samples) %*% x)
(bin.q <- quantile(rbinom(n, size = 800, mu), prob))

## 97.5%
## 762

percentiles <- c(lin.q, log.lin.q, pois.q, bin.q)

#Barplot the models with the predicted icecream units
plot_bar <- barplot(percentiles, names.arg = c("Linear", "Log-transformed",
"Poisson", "Binomial"),ylab="Predicted ice cream units", main="Predicted 97.5
percentile at 35 degree Centigrade")
text(plot_bar, percentiles-75, round(percentiles))
```

## Predicted 97.5 percentile at 35 degree Centigrade



*#The above plot displays the models and their predicted ice cream units. The value ranges from 762 to 2506. Since the market size was set to 800, the binomial model has a prediction of 762 units. The Log normal distribution is skewed to the right, hence has a prediction of 2506 units. The Poisson model is similar to the log-transform linear model having an exponential growth assumption.*

## **##Conclusion**

*#It is strongly believed that if the assumption of market size is 800, then binomial model should be considered.*